

**AFRL-IF-RS-TR-2005-303**  
**Final Technical Report**  
**August 2005**



# **WEBSRIPTER: END-USER TOOLS FOR COMPOSITION ONTOLOGY-ENABLED WEB SERVICES**

**University of Southern California at Marina del Rey**

**Sponsored by**  
**Defense Advanced Research Projects Agency**  
**DARPA Order No. K529**

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

**AIR FORCE RESEARCH LABORATORY**  
**INFORMATION DIRECTORATE**  
**ROME RESEARCH SITE**  
**ROME, NEW YORK**

## **STINFO FINAL REPORT**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-303 has been reviewed and is approved for publication

APPROVED:       /s/

JOSEPH A. CAROZZONI  
Project Engineer

FOR THE DIRECTOR:       /s/

JAMES W. CUSACK, Chief  
Information Systems Division  
Information Directorate

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved</i> <i>OMB No. 074-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> AUGUST 2005	<b>3. REPORT TYPE AND DATES COVERED</b> Final Jul 00 – Dec 04	
<b>4. TITLE AND SUBTITLE</b> WEBScripTER: END-USER TOOLS FOR COMPOSITION ONTOLOGY-ENABLED WEB SERVICES			<b>5. FUNDING NUMBERS</b> C - F30602-00-2-0576 PE - 62301E PR - DAML TA - 00 WU - 13	
<b>6. AUTHOR(S)</b> Martin Frank				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> University of Southern California Information Sciences Institute Marina del Ray California 90292			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  N/A	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> Defense Advanced Research Projects Agency AFRL/IFSB 3701 North Fairfax Drive 525 Brooks Road Arlington Virginia 22203-1714 Rome New York 13441-4505			<b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>  AFRL-IF-RS-TR-2005-303	
<b>11. SUPPLEMENTARY NOTES</b>  AFRL Project Engineer: Joseph A. Carozzoni/IFSB/(315) 330-7796/ Joseph.Carozzoni@rl.af.mil				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.				<b>12b. DISTRIBUTION CODE</b>
<b>13. ABSTRACT (Maximum 200 Words)</b> The World Wide Web (WWW) contains a large amount of information which is currently being represented using the Hypertext Markup Language (HTML) and the Extensible Markup Language (XML). However, HTML and XML have a limited capability to describe the relationships (schemes or ontologies) with respect to objects. The DARPA Agent Markup Language (DAML) through the use of ontologies provides a very powerful way to describe objects and their relationships to other objects. DAML is an extension to XML and the Resource Description Framework (RDF). This effort focused on developing WebScripTER to be an incentive for DAML usage for both information producers and information consumers by making it easy to build reports of existing DAML sources. In addition to generating a report of DAML sources, WebscripTER can pull information from multiple DAML sources and implicitly define relationships that can be used to align different ontologies.				
<b>14. SUBJECT TERMS</b> DAML, W3, RDF, OIL, HTML, XML				<b>15. NUMBER OF PAGES</b> 21
				<b>16. PRICE CODE</b>
<b>17. SECURITY CLASSIFICATION OF REPORT</b>  UNCLASSIFIED	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b>  UNCLASSIFIED	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b>  UNCLASSIFIED	<b>20. LIMITATION OF ABSTRACT</b>  UL	

# TABLE OF CONTENTS

Chapter 1: Introduction .....	1
1.1 What is DAML, OIL, the Semantic Web? .....	1
1.2 What is WebScripiter? .....	1
1.3 Why would I want to use WebScripiter? .....	1
1.4 What are examples of WebScripiter use? .....	2
1.5 How can I obtain WebScripiter? .....	2
Chapter 2: Installation .....	3
Chapter 3: Tutorial .....	4
3.1 Generating a simple WebScripiter report .....	4
Chapter 4: Reference .....	8
4.1 Single-ontology reports .....	9
4.2 A multiontology report .....	13
</Definition> Chapter 5: Background and Pointers .....	15
Chapter 5: Background and Pointers .....	16
5.1 XML .....	16
5.2 RDF .....	16
5.3 RDF Schema .....	16
5.4 DAML .....	16
Appendix A .....	17

# Chapter 1: Introduction

## ***1.1 What is DAML, OIL, the Semantic Web?***

DAML is the United States Defense Advanced Research Projects Agency's (DARPA) Agent Markup Language. It is an effort to define a Web-based machine-understandable knowledge representation. OIL, the Ontology Inference Layer, is a European sister effort. DAML+OIL merges the languages into a single proposed standard.

Both these efforts are part of an attempt to evolve the current World-Wide-Web (W3) into the Semantic Web in which machines can read content without “Web-scaping” or “wrapping” Web site content by hand.

<http://www.sciam.com/2001/0501issue/0501bernerslee.html> is an excellent general-audience introduction to the concept of the Semantic Web.

## ***1.2 What is WebScripter?***

WebScripter is a tool that allow ordinary users to make reports of DAML data distributed over the Web. You can add columns by browsing any DAML data in a hierarchy outline and then add data in a new column or to an existing column. While you are doing that, WebScripter records which ontological classes and fields were used together (in a DAML format), so that others can draw from that when making their reports (“people who used ISI:LastName also used Oxford:Surname and Karlsruhe:Vorname”). Thus, WebScripter also offers no-cost implicit collaborative ontology translation.

## ***1.3 Why would I want to use WebScripter?***

Using DAML and WebScripter, you can publish data once and then republish it many times. In a large organization, you can allow your individual units to publish their data in a format that makes sense to them, yet integrate it into seamless reports using WebScripter. You can break out a data entry task both “horizontally” (different units

supply different instances/rows), or “vertically” (different units supply different attributes/columns for the same instances), or any combination thereof.

### ***1.4 What are examples of WebScripter use?***

For example, point your browser to <http://www.isi.edu/divisions/div2> and click on People.

What you’re seeing is the output of a WebScripter report (albeit formatted with a custom XSL stylesheet). For example, the information about the office locations and phone numbers is maintained by the ISI business office while the matrix of who works on which projects is maintained by our Division, and the individual interests are maintained by individuals. WebScripter fuses all this information together into a coherent report.

Scroll all the way down on the page and click on “report” in the sentence “This page is automatically generated and maintained based on a Webscripter report that fetches and fuses...” and you click on that. What you are seeing is the standard look of a WebScripter report. The color coding indicates the source of the DAML data. The polished People page is made by a simple XSL transformation of the XML version of this WebScripter report.

### ***1.5 How can I obtain WebScripter?***

Go to <http://www.isi.edu/webscripter> and click on Download. This gives you everything you need to run WebScripter and make reports, including example reports, and this manual.

You can obtain the complete source code of WebScripter by signing a license agreement with us.

You can also use the WebScripter servlet service at to run WebScripter reports without installing any software. Go to <http://www.isi.edu/webscripter> and click on “as Web Service”.

## Chapter 2: Installation

The only requirement is a platform that supports the Sun Microsystem Java Development Kit (JDK) 1.4.0 or higher (note that we know for sure that JDK 1.3.1 will not work).

Say “`jar xvf webscripтерdistribution.jar`” if you have the JDK bin directory in your path, or something like “`C:/jdk1.4.0/bin/jar xvf webscripтерdistribution.jar`” otherwise. If you are on a PC, you can alternatively use WinZip to unpack the WebScripter distribution.

Either way you will end up with a directory called webscripтер on your hard disk. If you are on a PC and you have the JDK bin directory in your path you should be able to simply double-click on “`runinteractivewindows.bat`” in the webscripтер directory to start WebScripter. On a PC, you can also create a shortcut to that file and place the shortcut on your desktop as a convenience.

If you are running on a Macintosh or Unix variant, copy and modify the “`runinteractivewindows.bat`” file to follow the platform’s conventions (for example, Unix uses colons to separate CLASSPATH entries rather than the semicolons used by Windows).

## Chapter 3: Tutorial

This part of the manual describes how to use the graphical version of WebScripter. (The Reference chapter explains how to work with the textual XML format of a WebScripter report definition, which is more cumbersome but also more powerful, and can also be used to rerun reports you defined interactively from the command line.)

### ***3.1 Generating a simple WebScripter report***

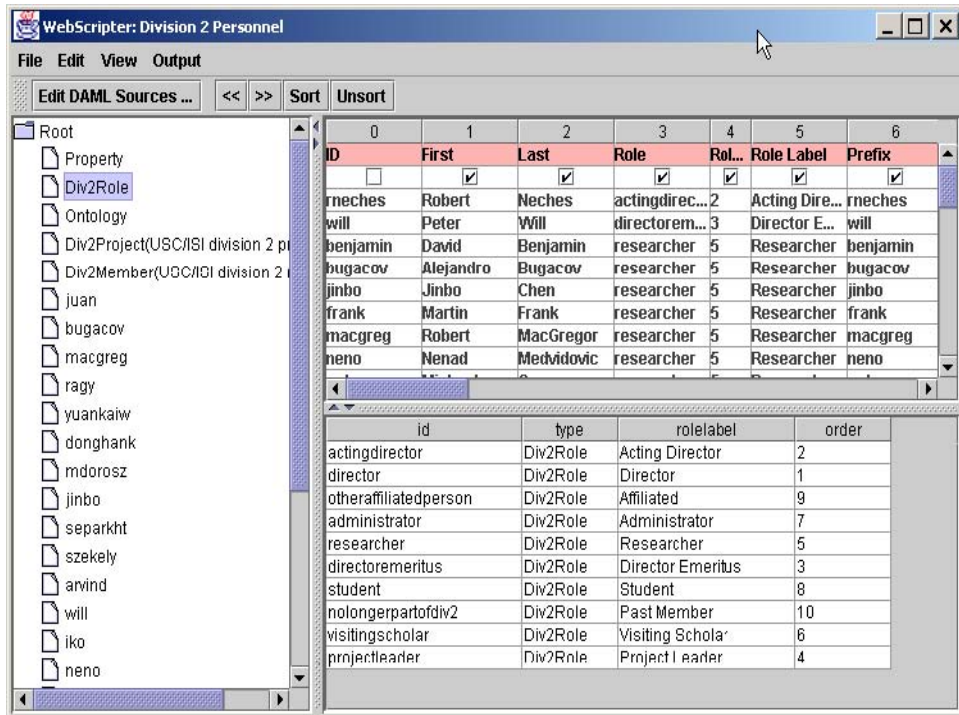
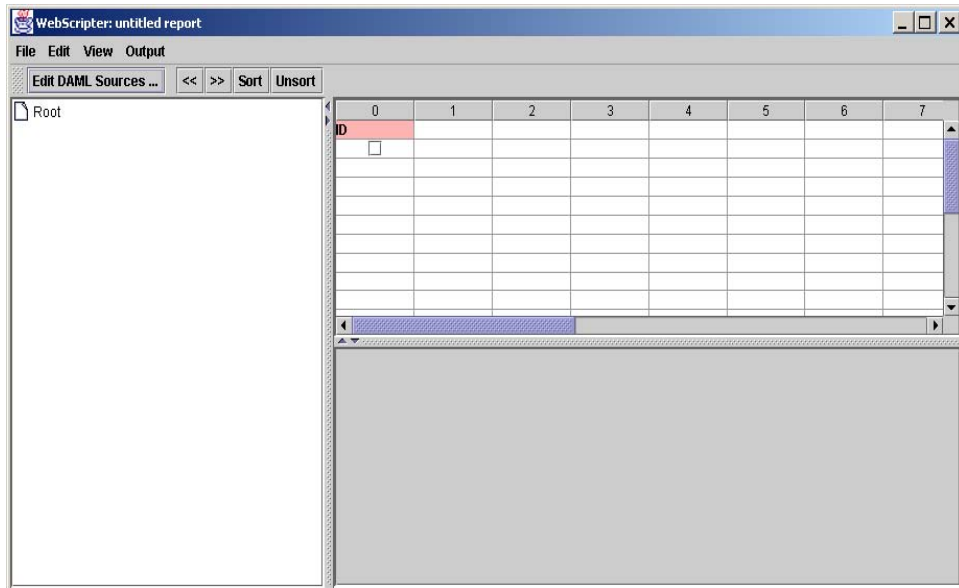
Start the graphical version of WebScripter as described in Chapter 2. You should see a window with a blank spreadsheet.

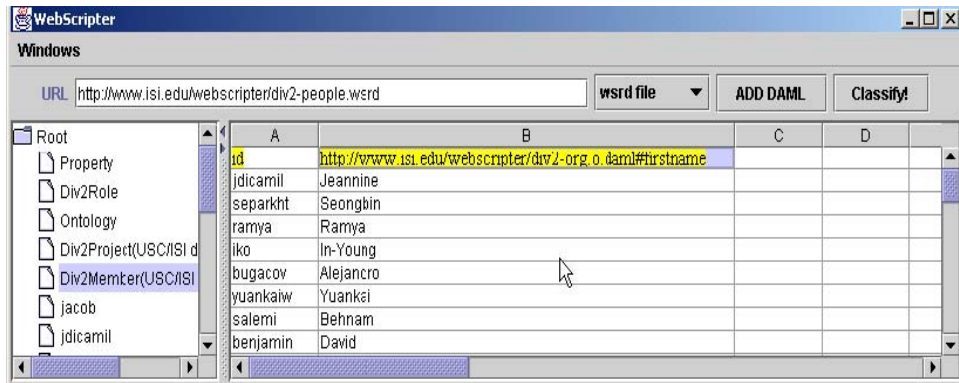
Load the URL <http://www.isi.edu/webscripter/div2people.gen.html> into your favorite Web browser. What you are looking at is the HTML output of a previous run of Web scripter. However, the content is also embedded as DAML into the Web page. (Say “View Source” in your Web browser and scroll towards the end.)

Thus, you can use the embedded DAML in that page for further processing. Choose “Example Report” from the file menu. You should see an outline of a simple concept hierarchy on the left, and the content of a report in the upper right. Left click on Div2Role.

Now, let’s be more selective about which data we want to appear in our report. Choose “New Report” from the File menu. Go to any field in the second or subsequent row of column B and type “Martin”. Highlight the field by first clicking on another cell and then clicking on the “Martin” cell (you should see a border around it), then click on the Classify button. Following are some example screen shots:







A popup window will appear which most likely will give you only a single choice which is “firstname”. Choose it and click on “Set”. You now have a report which lists our first names and nothing else. WebScripter has automatically classified the data you typed to the DAML attribute URI “http://www.isi.edu/webscripter/div2-org.o.daml#firstname”.

Classifying is just one way of adding columns, for one of a single ontological type it is actually easier to do this: click on Div2Member, highlight a field in the “firstname” column, and say choose “new report column” from the right mouse button menu for the same result.

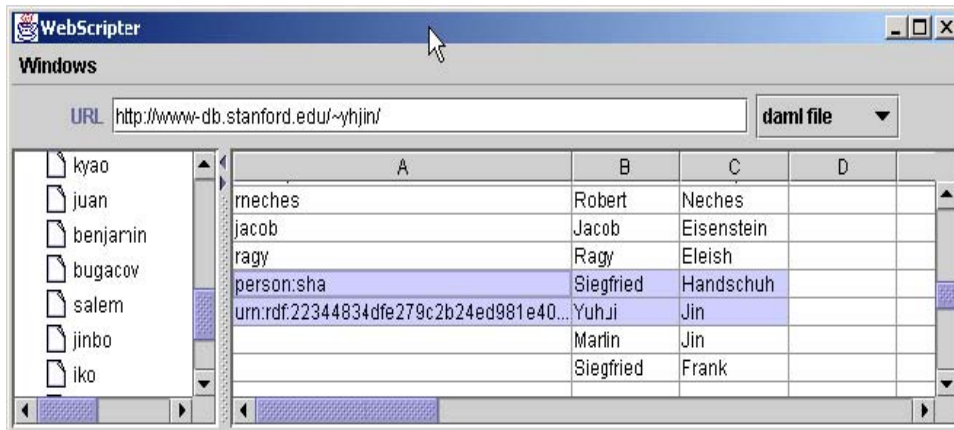
Scroll down and type “Frank” in column C in a blank row, select the cell, and click on Classify again. You will probably have a whole bunch of matches in the popup window now because “frank” is my ISI email prefix and used as a key for many attributes. Choose “lastname” from the choices given. Our last names now appear as well in column C.

So far, we have only drawn data from one data source, the Information Sciences Institute’s. What makes WebScripter and DAML in general exciting is that you can draw in and mix data from many different sources.

Load <http://www.aifb.unikarlsruhe.de/WBS/sha/index.html> as a DAML file. Load <http://www.wdb.stanford.edu/~yhjin/> as a DAML file. (You can find both of these people and more listed under <http://www.wdb.stanford.edu/Ontoagents/#participants>.)

Now type “Siegfried” in an empty cell of column B, select it, and Classify it; select “firstName” from the popup window and click “Set”. Type “Jin” in an empty cell of column C, select it, and Classify it; select “lastName”.

The below instructions rely on external data not maintained by us; so we can’t guarantee that the pages won’t have disappeared etc. by the time you read this.



Both Siegfried and Yuhui now appeared in the list of people's names.

Finally, click in the yellow header column, and you will see a popup window that lists the URIs of DAML properties that go into this column. You have also implicitly declared that you regard those two properties as equivalent at least for the purpose of this report; WebScripter will write out that fact as DAML "samePropertyAs" statements which can later benefit yourself and others by WebScripter offering to incorporate all of those whenever any of them are used. (More about that in a later chapter.)

Choose "Mapping Info" from the Output menu to view this equivalency information in DAML.

Choose "As HTML" from the Output menu to see the standard HTML format of the report you just created.

Choose "Report Definition" from the Output menu to obtain the WebScripter report definition "wsrd" file; it is in XML format, and you can refer to Chapter 4 to edit the XML by hand (for example, this lets you sort rows, restrict rows by content, and perform other operations that can currently not yet be performed interactively).

Finally, WebScripter's "Easy Publish" feature in the "Windows" menu will let you save your WebScripter report definition on an ISI server, and will return a URL which will dynamically recomputes the report content whenever anyone accesses it. If either you have no Internet connection or our servlet service is down WebScripter will go blank for 20 seconds or so, then return to life with a Java exception printed to standard error (which you can safely ignore).

Easy Publish gives you a quick way to play with WebScripter and share the results with others. You should obviously (a) not use the Easy Publish feature for production level content that is mission critical to your organization [we make no guarantees that the servlet will remain available forever], and (b) not use it to publish reports of content you want to keep secret [because anyone can see any Easy Publish report].

## Chapter 4: Reference

The interactive version of WebScripter is now a solid way of constructing and maintaining simple reports, but you can also edit report definitions manually, and then load them back into interactive WebScripter, or you can run a WebScripter report via the following command line if you are in the WebScripter distribution directory that contains all of the Jar files. (On a PC, on other platforms, you may have to replace the semicolons with colons and so forth.)

```
Java -classpath
'jena.jar;wsall.jar;rdf-api-2001-01-19.jar;xerces.jar;xalan.jar;gnu-regexp-1.0.8.jar'
Dorg.xml.sax.parser=org.apache.xerces.parsers.SAXParser
edu.isi.dce.webscripter.report.main.WsMnMain
-parser jena
-definition http://www.isi.edu/webscripter/div2people.wsrp
-printAllTriples false
-htmlOutput example.gen.html
-xmlOutput example.gen.xml
```

About the command line arguments: the parser argument is either jena or rdfapi, latter is only recommended for backward compatibility; -printAllTriples will do just that after loading all your DAML input examining that is the only sensible way to make WebScripter reports of pure RDF data with neither a DAML nor an RDFS schema behind it because it will not show you a class hierarchy on the left in that case.

WebScripter reports explicitly specify the DAML data source pages that will be loaded to produce the report. (Additional pages can also be loaded dynamically; more about that later.) In the future, explicitly specifying source pages may not always be necessary by using a web-crawler that knows where to find DAML content.

The data that results from instantiating a WebScripter report definition is always in tabular form. (Note that it is not in "relational" form in the database sense because it is not in First Normal Form – a single cell in the table can have multiple values.)

The first column of a table is special because it defines which rows will appear in the report; we call it the Generator Column. It is the seed of all other content in the table.

The second and all subsequent columns are Extractor Columns. They are computed from the cell content of a specified earlier column and a transformation. RDF triples have a subject predicate object structure; the cell content of an extractor column is computed by collecting all objects that match the column's predicate for the subjects in the source cell; this can also be reversed by setting IsInverse to true.

Sometimes you want to have certain columns in your report even though you don't

want to show then to the end users. For example, you can use a column for sorting or filtering the rows of the report but hide the column itself. You can do that for any column by setting the Visible tag to false.

You can sort by one or more columns, ascending or descending.

You can also filter out rows. Every cell in a WebScripter report can in principle have multiple values. Thus, filter expressions operate on list valued cells, even if you believe that there should always be at most one value for a cell. (But you should plan for the case that there are multiple values; for example, we initially expected to-do items to relate to exactly one project, but it later turned out that it made perfect sense that some to-do items such as Meta-Ja improvements apply to multiple projects.) Here are some simple examples.

```
<KeepRowIf>contains(*2002.*)</KeepRowIf>
<KeepRowIf>contains(http://www.isi.edu/~frank/metaja)</KeepRowIf>
<KeepRowIf>doesNotContain(\#ClosureReport)</KeepRowIf>
<KeepRowIf>doesNotContain(szekely|rneches)</KeepRowIf>
```

The expression within the brackets is a GNU regular expression, which means you can use sophisticated wild cards. A good overview of GNU regular expressions is at <http://www.emerson.emory.edu/services/editors/ne/Regular Expressions.html> (this is an external site which we have no control over try “GNU regular expression” in a search engine it does not work).

Finally, note that you can load from URLs which do not start with “http:” In particular, you can use “file:x.daml” to load local file x.daml, presuming such a file exists in your WebScripter directory (the one that contains wsall.jar ). You can probably also load from subfolders using “file:subfolder/x.daml” syntax and from superfolders using “file:../whatever/x.daml” syntax, but that is specific to your operating system and Java runtime implementation.

## 4.1 Single-ontology reports

Here is how I keep to-do item lists on the Meta-Ja project (<http://www.isi.edu/frank/metaja/>).

First, here is the list of open items. In a nutshell, the report lists all to-do items that (a) refer to the Meta-Ja project, and that (b) are not referred to by any closure report.

<http://www.isi.edu/webscripter/metajatodo.gen.html>

```
<Definition>
<ReportName>Meta-Ja Desiderata</ReportName> <Description>
```

These items describe potential enhancements to Meta-Ja that I believe are worthwhile in principle. The existence of items on this list does not imply that I promise to ever implement any of them not even if you have paid a commercial Meta-Ja licensee fee  
(because maintaining MetaJa is not part of my regular job the licensing fees are not nearly enough to cover my time).

```

</Description>
<GeneratorColumn>
  <Id>Id</Id>
  <Visible>true</Visible>
  <GeneratorSources>
    <GeneratorSource>
      <UrlOfAnRdfFile>http://www.isi.edu/webcrawler/todo.gen.d.daml</UrlOfAnRdfFile>
    </GeneratorSource>
    <GeneratorSource>
      <UrlOfAnRdfFile>http://www.isi.edu/webcrawler/mrftodoaddenda.d.daml</UrlOfAnRdfFile>
    </GeneratorSource>
  </GeneratorSources>
  <ClassIdentifiers><String>http://www.isi.edu/webcrawler/todo.o.daml#ToDoItem</String></ClassIdentifiers>
</GeneratorColumn>
<ExtractorColumns>
  <ExtractorColumn>
    <Id>ToDo Item</Id>
    <AnchorColumnId>Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webcrawler/todo.o.daml#label</Uri></Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Project</Id>
    <AnchorColumnId>Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webcrawler/todo.o.daml#project</Uri></Predicate></Predicates>
    <KeepRowIf>contains(http://www.isi.edu/~frank/metaja)</KeepRowIf>
    <Visible>false</Visible>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Priority</Id>
    <AnchorColumnId>Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webcrawler/todo.o.daml#priority</Uri></Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Description</Id>
    <AnchorColumnId>Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webcrawler/todo.o.daml#description</Uri></Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Entered</Id>
    <AnchorColumnId>Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webcrawler/todo.o.daml#entered</Uri></Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Annotations</Id>
    <Visible>false</Visible>
    <AnchorColumnId>Id</AnchorColumnId>
    <Predicates><Predicate>
      <Uri>http://www.isi.edu/webcrawler/todo.o.daml#todoitemid</Uri>
      <Inverse>true</Inverse>
    </Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Annotation Types</Id>
    <Visible>false</Visible>
    <AnchorColumnId>Annotations</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.w3.org/1999/02/22rdfsyntaxs#type</Uri></Predicate></Predicates>
    <KeepRowIf>
      doesNotContain(http://www.isi.edu/webcrawler/todo.o.daml#ClosureReport)
    </KeepRowIf>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Comment Dates</Id>
    <Visible>false</Visible>
    <AnchorColumnId>Annotations</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webcrawler/todo.o.daml#entered</Uri></Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Comments</Id>
    <Visible>false</Visible>
    <AnchorColumnId>Annotations</AnchorColumnId>

```

```

    <Predicates><Predicate><Uri>http://www.isi.edu/webscripter/todo.o.daml#content</Uri></Predicate></Predicates>
  </ExtractorColumn>
</ExtractorColumns>
<SortingInstructions>
  <SortingInstruction>
    <ColumnId>Entered</ColumnId>
    <Ascending>false</Ascending>
  </SortingInstruction> </SortingInstructions>
</Definition>

```

Second, here is the WebScripter report definition of closed items. It lists to do items that refer to the MetaJa project and that do have a closure report associated with them.

<http://www.isi.edu/webscripter/metajadone.gen.html>

```

<Definition>
<ReportName>MetaJa Enhancements</ReportName> <Description>
  Here is a list of what I recently did on MetaJa (listed by closure date, most recent first).
</Description>
  <GeneratorColumn>
    <Id>ToDo Id</Id>
    <Visible>false</Visible>
    <GeneratorSources>
      <GeneratorSource>
        <UriOfAnRdfFile>http://www.isi.edu/webscripter/todo.gen.d.daml</UriOfAnRdfFile>
      </GeneratorSource>
      <GeneratorSource>
        <UriOfAnRdfFile>http://www.isi.edu/webscripter/mrftodoaddenda.d.daml</UriOfAnRdfFile>
      </GeneratorSource> </GeneratorSources>
    <ClassIdentifiers><String>http://www.isi.edu/webscripter/todo.o.daml#ToDoItem</String></ClassIdentifiers>
  </GeneratorColumn>
<ExtractorColumns>
  <ExtractorColumn>
    <Id>ToDo Item</Id>
    <AnchorColumnId>ToDo Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webscripter/todo.o.daml#label</Uri></Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Project</Id> <AnchorColumnId>ToDo Id</AnchorColumnId> <Visible>false</Visible>
    <Predicates><Predicate><Uri>http://www.isi.edu/webscripter/todo.o.daml#project</Uri></Predicate></Predicates>
    <KeepRowIf>contains(http://www.isi.edu/~frank/metaja)</KeepRowIf>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Privacy Setting</Id> <AnchorColumnId>ToDo Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webscripter/todo.o.daml#privacysetting</Uri></Predicate></Predicates>
    <KeepRowIf>doesNotContain(. *private.*)</KeepRowIf> <Visible>false</Visible>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Description</Id> <AnchorColumnId>ToDo Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webscripter/todo.o.daml#description</Uri></Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Entered</Id> <AnchorColumnId>ToDo Id</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.isi.edu/webscripter/todo.o.daml#entered</Uri></Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Annotations</Id>
    <Visible>false</Visible>
    <AnchorColumnId>ToDo Id</AnchorColumnId>
    <Predicates><Predicate>
      <Uri>http://www.isi.edu/webscripter/todo.o.daml#todoitemid</Uri>
      <Inverse>true</Inverse>
    </Predicate></Predicates>
  </ExtractorColumn>
  <ExtractorColumn>
    <Id>Annotation Types</Id>
    <Visible>false</Visible>
    <AnchorColumnId>Annotations</AnchorColumnId>
    <Predicates><Predicate><Uri>http://www.w3.org/1999/02/22rdfsyntaxns#type</Uri></Predicate></Predicates>
    <KeepRowIf>
      contains(http://www.isi.edu/webscripter/todo.o.daml#ClosureReport)
    </KeepRowIf>
  </ExtractorColumn>

```

```

</ExtractorColumn>
<ExtractorColumn>
  <Id>Closed</Id>
  <AnchorColumnId>Annotations</AnchorColumnId>
  <Predicates><Predicate><Uri>http://www.isi.edu/webscripiter/todo.o.daml#entered</Uri></Predicate></Predicates>
</ExtractorColumn>
<ExtractorColumn> <Id>Closure Remark</Id> <AnchorColumnId>Annotations</AnchorColumnId>
  <Predicates><Predicate><Uri>http://www.isi.edu/webscripiter/todo.o.daml#remark</Uri></Predicate></Predicates>
</ExtractorColumn>
</ExtractorColumns>
<SortingInstructions>
<SortingInstruction>
<ColumnId>Entered</ColumnId>
<Ascending>false</Ascending>
</SortingInstruction> </SortingInstructions>
</Definition>

```



## 4.2 A multiontology report

The report below is interesting because it draws data from multiple DAMLized Web pages that use multiple ontologies and whose authors did not consult each other when marking up their pages.

<http://www.isi.edu/webscripter/damlpersonnel.gen.html>

```
<Definition>
<ReportName>DAML Personnel</ReportName> <Description>
  An attempt to pull together as much information about DAML personnel as I
  (Martin Frank) can find data for, using WebScripter.
</Description>
<GeneratorColumn>
<Id>Id</Id>
<Visible>true</Visible>
<GeneratorSources>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.db.stanford.edu/Ontoagents/#participants</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.db.stanford.edu/~stefan/#person_stefan_decker</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.aifb.unikarlsruhe.de/WBS/sha/index.html#person:sha</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.db.stanford.edu/~melnik/index.daml</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.db.stanford.edu/~yhjin</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.aifb.unikarlsruhe.de/~sst</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.ksl.stanford.edu/people/dlm/</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://kslweb.stanford.edu/people/sam/</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.cs.umd.edu/~hendler/jhendler.daml</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.cs.cmu.edu/~softagents/atlas/daml_pages/payne_terence.daml</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource> <UrlOfAnRdfFile>http://www.semanticweb.org/ontologies/swrc-
  onto-2000-09-10.daml</UrlOfAnRdfFile> </GeneratorSource>

  <GeneratorSource>
    <UrlOfAnRdfFile>http://www.isi.edu/webscripter/div2org.o.daml</UrlOfAnRdfFile>
  </GeneratorSource>
  <GeneratorSource>
    <UrlOfAnRdfFile>http://www.isi.edu/webscripter/div2roster.d.daml</UrlOfAnRdfFile>
  </GeneratorSource>
  <GeneratorSource>
    <UrlOfAnRdfFile>http://www.isi.edu/webscripter/div2projects.gen.d.daml</UrlOfAnRdfFile>
  </GeneratorSource>
  <GeneratorSource>
    <UrlOfAnRdfFile>http://www.isi.edu/webscripter/div2roles.d.daml</UrlOfAnRdfFile>
  </GeneratorSource>
  <GeneratorSource>
    <CrawlDepth>1</CrawlDepth>
    <UrlOfAnRdfFile>http://www.isi.edu/webscripter/div2registry.d.daml</UrlOfAnRdfFile>
  </GeneratorSource>
</GeneratorSource>
```

```

<UrlOfAnRdfFile>http://www.isi.edu/webcrawler/div2display.d.daml</UrlOfAnRdfFile>
</GeneratorSource>
<GeneratorSource>
  <UrlOfAnRdfFile>http://www.isi.edu/webcrawler/damlpersonneladdenda.daml</UrlOfAnRdfFile>
</GeneratorSource>
</GeneratorSources>
<ClassIdentifiers>
<String>http://www.isi.edu/webcrawler/div2org.o.daml#Div2Member</String>
  <String>http://www.semanticweb.org/ontologies/swrconto20000910.daml#Person</String>
  <String>http://www.semanticweb.org/ontologies/swrconto20000910.daml#FullProfessor</String>
  <String>http://www.w3.org/1999/02/22rdfsyntaxns#FacultyMember</String>
  <String>http://www.semanticweb.org/ontologies/swrconto20000910.daml#PhDStudent</String>
  <prevent duplicate entry for Stefan http://www.semanticweb.org/ontologies/swrconto20000910.daml#AcademicStaff
  >
  <String>http://www.semanticweb.org/ontologies/swrconto20000910.daml#Lecturer</String>

  <String>http://www.ksl.stanford.edu/projects/DAML/ksldamldesc.daml#Person</String>
  <String>http://ksl.stanford.edu/Projects/DAML/ksldamldesc.daml#PERSON</String>
  <String>http://ksl.stanford.edu/Projects/DAML/ksldamldesc.daml#PERSON</String>

</ClassIdentifiers>
</GeneratorColumn>
<ExtractorColumns>
<ExtractorColumn>
  <Id>Name</Id>
  <AnchorColumnId>Id</AnchorColumnId>
  <Predicates>
    <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#name</Uri></Predicate>
    <Predicate><Uri>http://www.ksl.stanford.edu/projects/DAML/ksldamldesc.daml#hasFullName</Uri></Predicate>
    <Predicate><Uri>HasFullName</Uri></Predicate>
  </Predicates>
</ExtractorColumn>
<ExtractorColumn>
<Id>First</Id>
  <AnchorColumnId>Id</AnchorColumnId>
  <Predicates>
    <Predicate><Uri>http://www.isi.edu/webcrawler/div2org.o.daml#firstname</Uri></Predicate>
    <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#firstName</Uri></Predicate>
  </Predicates>
<KeepRowIf>doesNotContain(Jeannine)</KeepRowIf>
</ExtractorColumn>
<ExtractorColumn>
<Id>Last</Id>
  <AnchorColumnId>Id</AnchorColumnId>
  <Predicates>
    <Predicate><Uri>http://www.isi.edu/webcrawler/div2org.o.daml#lastname</Uri></Predicate>
    <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#lastName</Uri></Predicate>
  </Predicates>
<KeepRowIf>doesNotContain(Park|Balasundaram|Ko|Bugacov|Wang|Salemi|Benjamin|Will|Yao|Seshadri|Kim|Chen|
Medvidovic|MacGregor|Eisenstein|Eleish)</KeepRowIf> </ExtractorColumn> <ExtractorColumn>
  <Id>Affiliation</Id>
  <AnchorColumnId>Id</AnchorColumnId>
  <Predicates>
    <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#affiliation</Uri></Predicate>
  </Predicates>
</ExtractorColumn>
<ExtractorColumn>
<Id>Role</Id>
  <AnchorColumnId>Id</AnchorColumnId>
  <Predicates>
    <Predicate><Uri>http://www.isi.edu/webcrawler/div2org.o.daml#role</Uri></Predicate>
    <Predicate><Uri>HasResearchPosition</Uri></Predicate>
  </Predicates>
</ExtractorColumn>
<ExtractorColumn>
<Id>Email</Id>
  <AnchorColumnId>Id</AnchorColumnId>
  <Predicates>
    <Predicate><Uri>http://www.isi.edu/webcrawler/div2org.o.daml#preferredEmailAddress</Uri></Predicate>
    <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#email</Uri></Predicate>
  </Predicates>
</ExtractorColumn>

```

```

        <Predicate><Uri>http://www.cs.umd.edu/projects/plus/DAML/onts/personal1.0.daml#emailAddress</Uri></Predicate>
      </Predicates>
    </ExtractorColumn>
    <ExtractorColumn>
      <Id>Phone</Id>
      <AnchorColumnId>Id</AnchorColumnId>
      <Predicates>
        <Predicate><Uri>http://www.isi.edu/webscripiter/div2org.o.daml#phonenumner</Uri></Predicate>
        <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#phone</Uri></Predicate>
        <Predicate><Uri>HasPhoneNumber</Uri></Predicate>
      </Predicates>
    </ExtractorColumn>
    <ExtractorColumn>
      <Id>Interests</Id>
      <AnchorColumnId>Id</AnchorColumnId>
      <Predicates>
        <Predicate><Uri>http://www.isi.edu/webscripiter/div2org.o.daml#interests20words</Uri></Predicate>
      </Predicates>
    </ExtractorColumn>
    <ExtractorColumn>
      <Id>Home Page</Id>
      <AnchorColumnId>Id</AnchorColumnId>
      <Predicates>
        <Predicate><Uri>http://www.isi.edu/webscripiter/div2org.o.daml#homepage</Uri></Predicate>
        <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#homepage</Uri></Predicate>
        <Predicate><Uri>HasHomePage</Uri></Predicate>
      </Predicates>
    </ExtractorColumn>
    <ExtractorColumn>
      <Id>Project IDs</Id>
      <AnchorColumnId>Id</AnchorColumnId>
      <Predicates>
        <Predicate><Uri>http://www.isi.edu/webscripiter/div2org.o.daml#workson</Uri></Predicate>
        <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto200009-10.daml#worksAtProject</Uri></Predicate>
        <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#headOf</Uri></Predicate>
        <Predicate><Uri>http://www.cs.umd.edu/projects/plus/DAML/onts/general1.0.daml#head</Uri></Predicate>
      </Predicates>
    </ExtractorColumn>
    <ExtractorColumn>
      <Id>Project Nicknames</Id>
      <AnchorColumnId>Project IDs</AnchorColumnId>
      <Predicates>
        <Predicate><Uri>http://www.w3.org/2000/01/rdfschema#label</Uri></Predicate>
        <Predicate><Uri>http://www.semanticweb.org/ontologies/swrconto20000910.daml#title</Uri></Predicate>
      </Predicates>
    </ExtractorColumn>
  </ExtractorColumns>
  <SortingInstructions>
    <SortingInstruction>
      <ColumnId>Last</ColumnId>
      <Ascending>true</Ascending>
    </SortingInstruction>
  </SortingInstructions>

```

</Definition>

## Chapter 5: Background and Pointers

Here are the normative references to the languages that WebScripiter uses.

### **5.1 XML**

<http://www.w3.org/XML/>

### **5.2 RDF**

<http://www.w3.org/RDF/>

### **5.3 RDF Schema**

<http://www.w3.org/TR/2000/CRrdfschema20000327/>

### **5.4 DAML**

<http://www.daml.org/2001/03/daml+oilindex.html>

# Appendix A

## Revision History

From 1.1 July 10th to 1.2 July 25th, 2001: added installation instructions for binary distribution

- 1.3 August 16, 2001: added more examples and generally cleaned up in preparation for first publicly downloadable release
- 2.0 February 9th, 2002: complete rewrite of internals, based on Jena DAML parser by default now, 12 orders of magnitude faster report processing, can handle 10+ MB of DAML input files, you no longer have to list subclasses explicitly to pull them in, GUI enhancements: sorting by multiple columns, moving columns, adding/deleting/modifying column definitions and DAML input file list, loading with crawl depth of 1 from the GUI
- 2.2 May 31st, 2002: large number of improvements, including guessing ok linkages between classes and offering a menu of that to the user; also sorting and ordering of the class hierarchy by various criteria; plain HTML pages can now be used as a jumpoff point for loading RDF pages it points to (ISWC'2002 support); can now parse embedded RDF out of HTML pages with Jena